Recommendation Systems: Comparison Analysis between Traditional Techniques and Graph Based Techniques

Greshma Babu
Wanci He
Raguwansh Raj
Vivian Koutroumani

01.

Introduction

Recommendation Systems

02.

Problem Definition

Methods e-commerce platforms use and their drawbacks, related work

03.

The dataset

Dataset selection, Data analysis - exploration

04.

Traditional methods

Collaborative filtering, Popularity based techniques, Matrix factorization, Market Basket Analysis

05.

Graph Based methods

Random Walk, Neural Embedding, Bipartitte Graph

06.

Evaluation - Conclusion

Metrics and results

Introduction

E-commerce platforms, offer a massive variety of products and information. In order to help the user to find out information about the product , recommendation systems where developed.

Recommendation systems are able to suggest items to users. The suggested items should be the most “relevant” possible item for the user, so that the user would be more likely to choose those items: Amazon products, news articles, Youtube videos etc.

Problem definition

Traditional methods

- Popularity based recommendation
- Classification model based
- Content based recommendations
- Collaborative Filtering: Two types:1) User-based and 2) Item based.
- Hybrid Approaches: This system approach is to combine collaborative filtering, content-based filtering, and other approaches .
- Association rule mining

Problems to be solved

- Difficulty to continuously update these massive datasets
- Complex computation of similarity among items (for content based approaches)

Problem definition - Our objective

- Graph Learning based Recommendation Systems have been developed on graphs, where the important objects, e.g., customers, products, and attributes, are either explicitly or implicitly connected.
- In our research, we explored graph-based recommendation systems using neural embedding to capture similarity in a sparse dataset.
- We used neural embedding to predict the co-purchase bipartite graph of items and baskets of customers.

Problem definition - Related work

We got mainly inspired by a paper we read (attached in the report) from Stanford University.

In their paper, Se Won Jang, Simon Kim and JeongWoo Ha compare collaborative filtering, random walk and Neural Embeddings Candidate Generation.

We also referred to the the paper's references, containing:
Li's paper, in which he's using a bipartite approach,
Mikolov's paper, who uses a word2vec model, and
F. Videla-Cavieres' and Sebastian A. Rio's research, in which they used graph mining techniques, to perform market basket analysis.

The Dataset

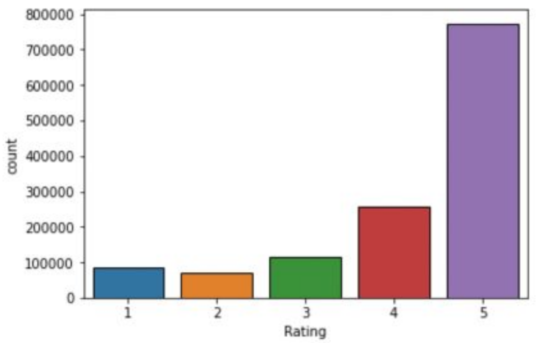
Amazon Reviews

For our project, we chose a dataset containing Amazon reviews for music and books.

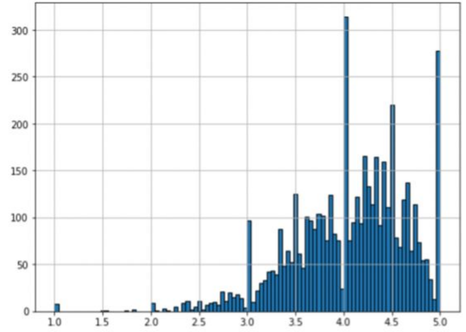
We used samples of the original dataset to train and test our algorithms. On the samples, we filtered out products that have less than a threshold of reviews, and users that have reviewed less than a number products, due to computer capacity limitations.

The Dataset - Data exploration

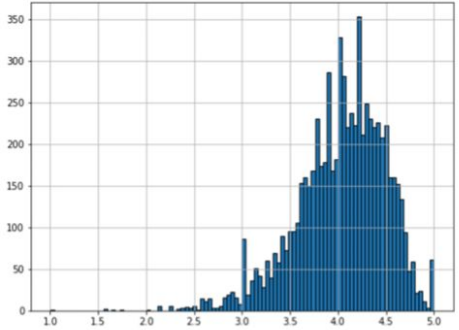
Our exploratory analysis showed that ratings were favorable in their majority.



There were outliers of course, but the median number of rating per user was 14-20, and the median number of ratings per product was 25.

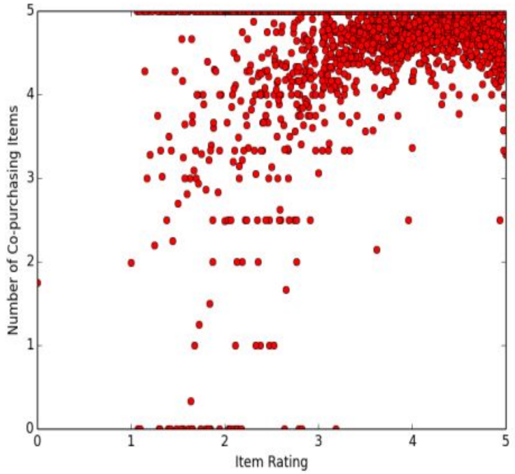


Music reviews distribution



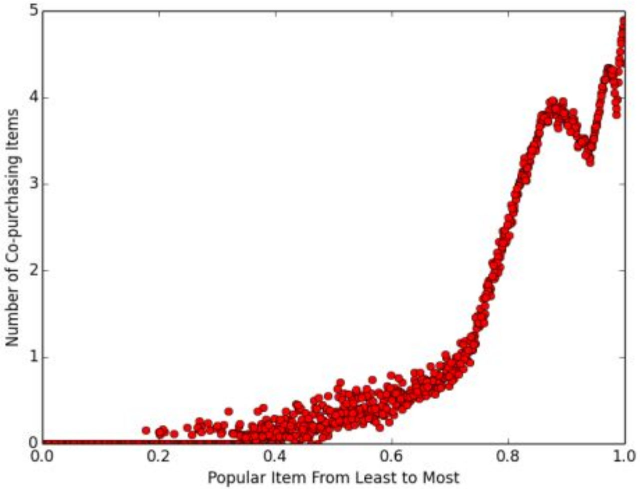
Book reviews distribution

The Dataset - Data Exploration

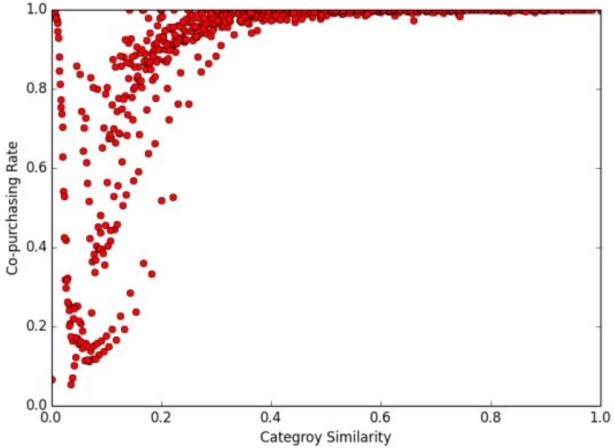


Item rating: weak Correlation
Searching for correlations, we see that generally, the higher ranked products have more co-purchased products

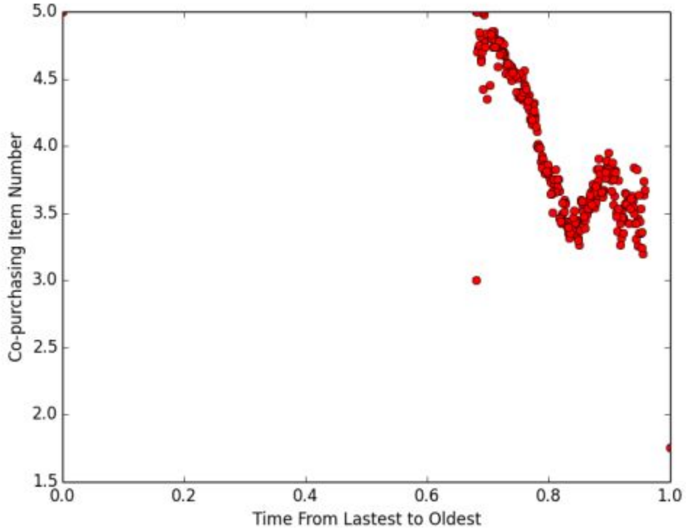
Sales Rank:High Correlation
Sales rank analysis: Correlation between the popularity of a product and the number of co-purchasing products the product has



The Dataset - Data Exploration



Category similarity:moderate correlation
Two items that belong to similar categories tend to be purchased together more frequently.
Category Similarity



TIme of first review : negative or low correlation
Time of first review analysis tries to see if there is a correlation between the time the product is first reviewed and the number of co-purchasing products the product has

Traditional Methods

1. **Collaborative filtering**
2. **Popularity based technique**
3. **Matrix factorization**
4. **Market Basket Analysis**

Collaborative filtering

Amazon uses item-item collaborative filtering which produces high quality recommendation system in the real time.

It aims to fill in the missing entries of a user-item association matrix, based on the idea that the best recommendations come from people who have similar tastes.

We used the kNNWithMeans algorithm, which is a basic collaborative filtering algorithm, considering the mean ratings of each user, 5 neighbors for aggregation and the pearson similarity.

We got an RMSE of 1.66 for the music test set, and 0.77 on the book test set.

Popularity based recommendation

In our approach, first we filtered out users that have given less than 50 reviews. For the remaining reviews, we counter the sum of ratings for each product, and the number of reviews per product, which works as a score to pick the top 5 recommendations.

Comparing the recommendations' average rating, with the actual ratings user have given to these products, we got an RMSE of 1.104 for books, and 1,217 for music.

Popularity based recommendation basically uses the items which are in trend right now.

It's not personalized so we don't take advantage of the individuals' behaviors.

Matrix factorization

CF method which focuses on the relationship between items and users.

With the input of users’ ratings on the products, we would like to predict how the users would rate the items so the users can get the recommendation based on the prediction. Given a ranking matrix of users and items, we want to fill the “gaps”

We used the TuriCreate Factorization Recommender, that took into account 8 factors of the books (and music respectively) and optimized the recommendations using the Stochastic Gradient Descent to minimize RMSE.

The final RMSE on the test set recommendations was 0.91

Market Basket Analysis

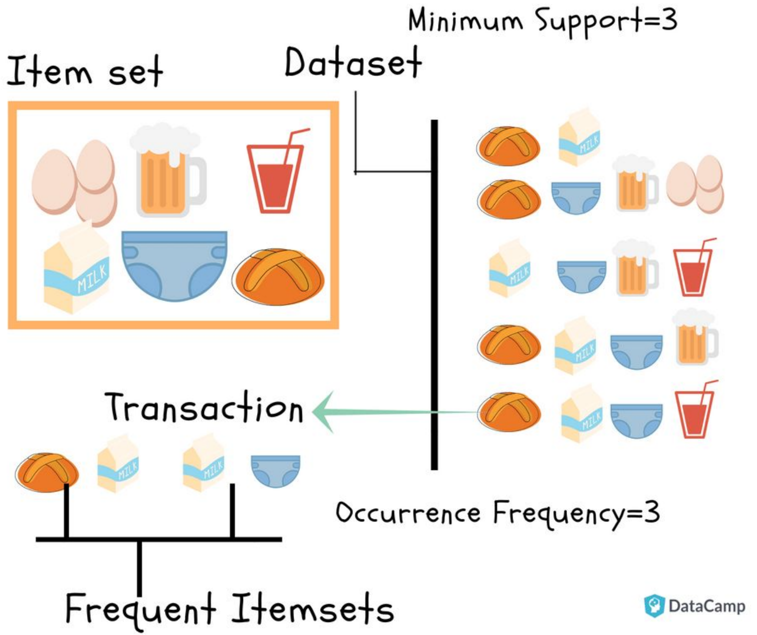
Market Basket Analysis looks at the different shopping baskets and checks for co-occurrences of products and constructs association rules for product pairs based on some parameters like support, confidence and lift.

$$Rule: \ X \Rightarrow Y$$

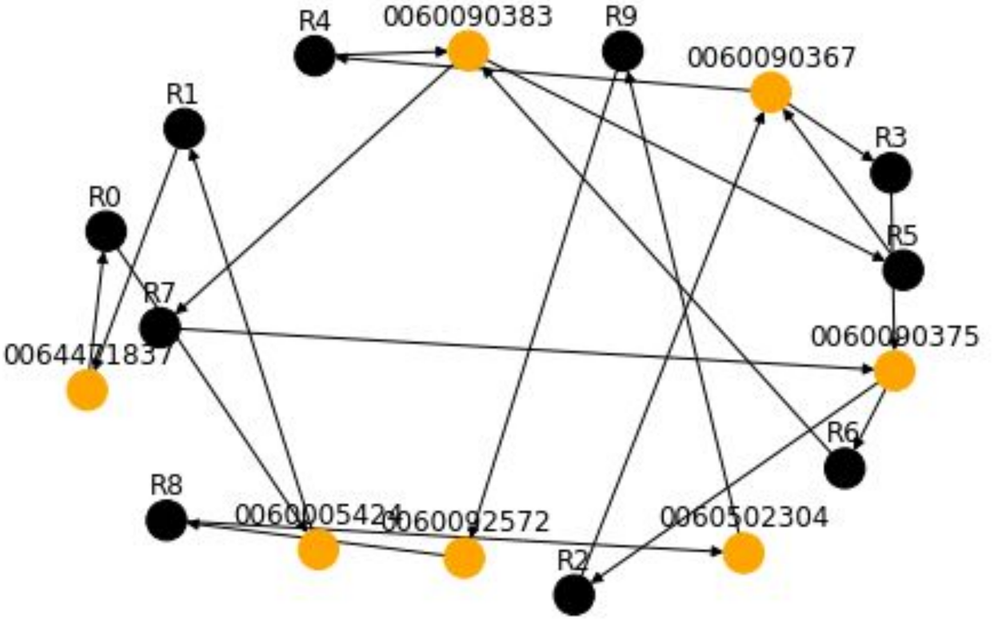
$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$



Network Graph for Association Rules



Graph based methods

1. Random Walk
2. Neural Embedding
3. Bipartite Graph based method

Random walk Method

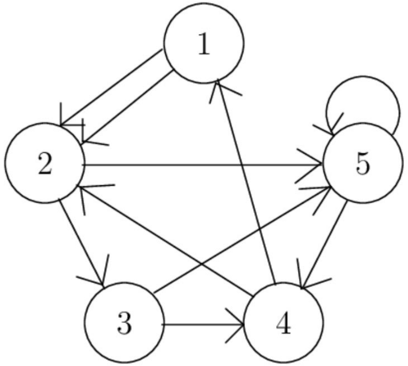


Figure 1.

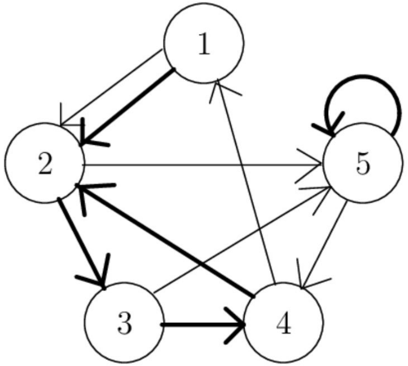


Figure 2.

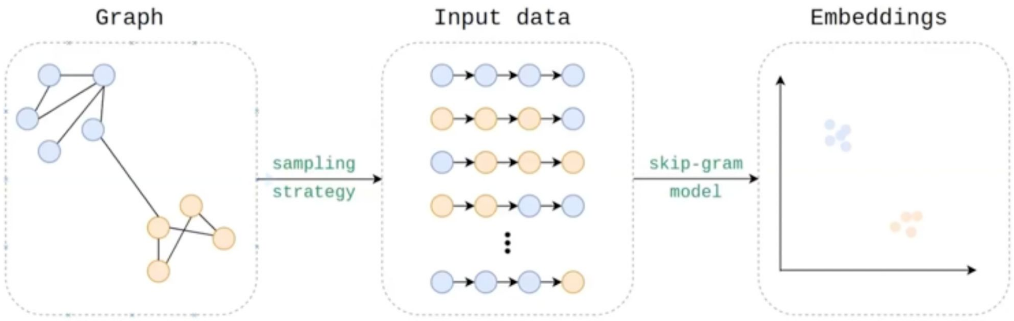
- This is a baseline model
- In random walk method, we traverse the path from each node randomly choosing one of their neighbours to reach a final node (here, after 10 steps).
- We repeat this process for 10 runs and so have 10 random walk destinations for each node
- We then compute the Jaccard similarity of this set of nodes
- The idea is that similar nodes would have more overlap in the final destinations

Evaluation of Random Walk

The path length between our product and the recommendation would of course depend on the hyperparameter.

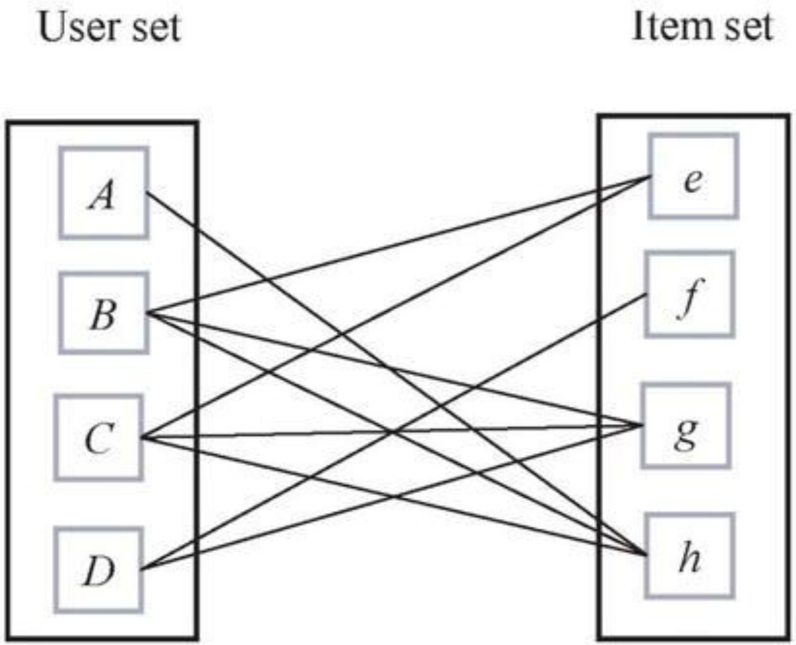
Here, we used a value of 10 steps (10 runs) with teleportation probability of 0.4. We obtained a average path length of 1.94 between each node and the recommendation when done for a sample.

Neural Embedding



- Sampling using DeepWalk, accompanied with negative sampling
- Train embeddings using skip-gram algorithms
- Optimize based on SGD
- Calculating cosine distance between embeddings, and recommend by selecting the top 5 similar products for each item
- The average path length is 2.3

Bipartite Graph



- The product nodes and user nodes are connected with odd length assuming no edge exist between same set
- Edge weights are obtained using text similarity between the nodes
- Number of iterations set to 10
- Global similarity is obtained which is very effective in sparse data
- personal rank to compute node similarity
- Average distance in co purchase graph for item and its recommendations was 2.8

Conclusion

1. Traditional methods like CF can recommend items with which no co-purchases have been made but however fails to get good similarity in bigger database - high memory consumption and exponential computation time.
Huge sparsity - > Lower performance

2. Graph model is very efficient in such cases as they are able to capture the embedding information more effectively.

3. We found that text similarity is really significant to predict co-purchases.

4. Further Improvement achievable with Graph Neural network to get the embedding for a given product, as they are able to capture direct and hidden similarity based on the structure of the graph and hence we can compute a more robust similarity between the user-item pair

Thank you for your attention!